Supplementary Material for

Remote sensing of chlorophyll-a in clear vs. turbid waters in large lakes

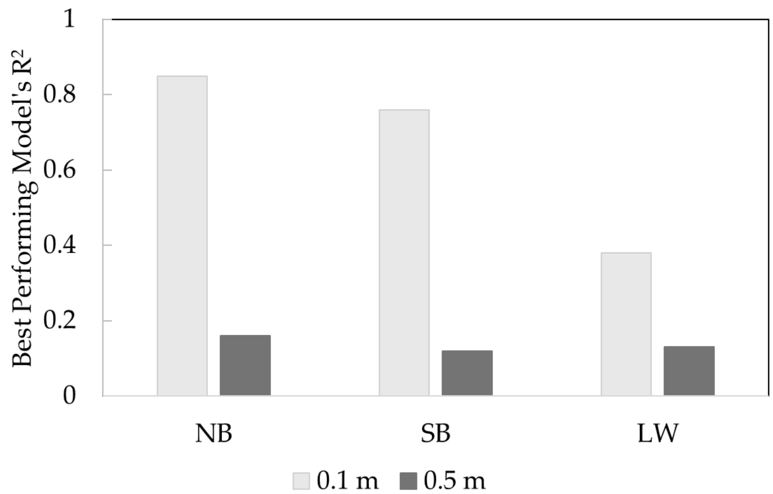

Figure S1. Best performing models' R^2 using *in situ* matchups from different depth are shown as light grey (0.1 m) and dark grey bars (0.5 m). Increase in the depth of *in situ* matchups from 0.1 to 0.5m reduces R^2 to about 5 times.

Table S1. Best calibration results for each tested Chl-*a* retrieval model. Values show the highest R^2 for each Chl-*a* retrieval model among the 12 sets of matchups for different temporal (0 to ±3 days) and spatial windows (1×1, 3×3, and 5×5 pixels) for NB-specific and SB-specific, and LW-specific matchups. Calibrations with no significant correlation between in situ and modeled chl-*a* ($p > 0.05$) were shown as dash.

Band math	NB-Specific	SB-Specific	LW-Specific
B[1]	-	-	-
G[2]	0.24	-	-
R[3]	-	0.26	-
NIR	-	0.21	-
B/G[4]	0.82	-	0.33
B/R[6=5]	0.24	0.45	-
B/NIR	-	0.24	-
G/B[6]	0.85	-	0.38
G/R[7]	-	0.75	-
G/NIR	-	0.20	0.25
R/B[8]	0.29	0.42	-
R/G[9]	-	0.76	-
R/NIR[10]	-	-	-
NIR/B[11]	-	0.25	-
NIR/G	-	0.22	0.24
NIR/R[12]	-	-	-
B×G	-	-	-
B×R	-	-	-
B×NIR	-	-	-
G×R	0.18	-	-
G×NIR	-	-	-
R×NIR	-	0.26	-
Avg(R;NIR)	-	0.25	-
B×R×NIR	-	0.20	-
Ave(G;R)	0.19	-	-
Ave(B;R)	-	-	-
(R/B)×NIR	-	0.33	-
(G/R)×NIR	-	-	-
(R/G)×NIR	-	0.32	-
(B×G×R)	-	-	-
(B×G×NIR)	-	-	-
(G×R×NIR)	-	0.22	-

Band math	NB-Specific	SB-Specific	LW-Specific
Ave(B;G)	-	-	-
Ave(B;NIR)	-	-	-
Ave(G;NIR)	-	-	-
(B/R)×NIR	-	-	-
((1/B)-(1/G))×NIR	0.82	-	0.29
((1/R)-(1/G))×NIR	-	0.74	0.27
((1/R)-(1/B))×NIR	0.26	0.43	-
NDVI [13]	-	-	-
NRVI[14]	-	-	-
(B-R)×G	0.44	0.45	-
SABI[15]	-	-	-
KIVU [16]	0.43	0.44	-
Kab1[17]	-	-	0.37

(1) Ritchie et al (1990); (2) Lathrop & Lillesand (1986); Östlund et al (2001); (3) Tyler et al (2006); Allan et al (2011); Allan et al (2015); (4) Sudheer et al (2006); (5) Han & Jordan (2005); (6) Doxaran et al (2009); (7) Hellweger et al (2004); (8) Sass et al (2007); (9) Floricioiu et al (2004); (10) Strmbech et al (2004); (11) Tebbs et al (2013); (12) Duan et al (2007); (13) Normalized Difference Vegetation Index (NDVI) = ((NIR-R)/(NIR+R)) in Mishra & Mishra (2012); (14) Normalized Ratio Vegetation Index (NRVI) = (((R/NIR)-1)/((R/NIR)+1)) in Zhengjun et al (2008); (15) Surface Algal Bloom Index (SABI) = (NIR-R)/(B+G) in Alawadi (2010); (16) 3BDA-like (KIVU) = (B-R)/G in Brivio et al (2001); (17) Kab1 = 1.67-3.94×ln(B)+3.78×ln(G) in Kabbara et al (2008).

Table S2. Calibration and validation results for the NB-scale Best Performing Models (BPMs) for different spatial and temporal windows. The values are average of the 5 cross validation folds. Calibrations with no significant correlation between *in situ* and modeled chl-*a* ($p > 0.05$) were shown as dash.

		Model	**Calibration R^2**	**RMSE (μgL^{-1})**	**RMSLE (μgL^{-1})**	**NRMSE**	**MAE (μgL^{-1})**	**MAPE (%)**	**Bias**
1 × 1	0 day	-	-	-	-	-	-	-	-
1 × 1	±1 day	G/B	0.85	31.94	0.74	1.98	22.81	67.72	9.39
1 × 1	±2 days	$(B^{-1}-G^{-1})\times NIR$	0.75	20.86	0.61	0.73	13.57	50.44	2.04
1 × 1	±3 days	$(B^{-1}-G^{-1})\times NIR$	0.57	23.37	0.72	0.88	14.48	38.68	7.25
3 × 3	0 day	-	-	-	-	-	-	-	-
3 × 3	±1 day	G/B	0.70	46.11	0.99	3.11	32.66	115.39	9.81
3 × 3	±2 days	G/B	0.66	23.93	0.72	0.71	16.49	43.11	0.37
3 × 3	±3 days	$(B^{-1}-G^{-1})\times NIR$	0.61	33.41	0.75	1.01	21.72	34.21	-3.89
5 × 5	0 day	-	-	-	-	-	-	-	-
5 × 5	±1 day	G/B	0.60	67.06	1.26	6.66	48.02	139.09	-5.86
5 × 5	±2 days	$(B^{-1}-G^{-1})\times NIR$	0.58	39.81	0.78	0.95	26.06	65.33	-8.69
5 × 5	±3 days	$(B^{-1}-G^{-1})\times NIR$	0.55	34.30	0.78	1.02	19.10	36.12	-1.35

Table S3. Calibration and validation results for the SB-scale Best Performing Models (BPMs) for different spatial and temporal windows. The values are average of the 5 cross validation folds. Calibrations with no significant correlation between *in situ* and modeled chl-*a* ($p > 0.05$) were shown as dash.

		Model	**Calibration R^2**	**RMSE (μgL^{-1})**	**RMSLE (μgL^{-1})**	**NRMSE**	**MAE (μgL^{-1})**	**MAPE (%)**	**Bias**
1 × 1	0 day	-	-	-	-	-	-	-	-
	±1 day	R/G	0.76	0.99	0.17	0.23	0.91	20.01	0.25
	±2 days	R/G	0.62	1.14	0.20	0.24	1	22.81	0.23
	±3 days	$(R^{-1}-G^{-1})\times NIR$	0.32	-	-	-	-	-	-
3 × 3	0 day	-	-	-	-	-	-	-	-
	±1 day	$(R^{-1}-G^{-1})\times NIR$	0.74	0.59	0.10	0.12	0.59	13.56	0.44
	±2 days	R/G	0.51	1.22	0.21	0.27	1.02	18.55	-0.08
	±3 days	-	-	-	-	-	-	-	-
5 × 5	0 day	-	-	-	-	-	-	-	-
	±1 day	R/G	0.63	2.35	0.37	0.35	2.35	54.30	2.35
	±2 days	R/G	0.63	1.20	0.20	0.26	1.02	19.75	1
	±3 days	-	-	-	-	-	-	-	-

Table S4. Calibration and validation results for the LW-scale Best Performing Models (BPMs) for different spatial and temporal windows. The values are average of the 5 cross validation folds. Calibrations with no significant correlation between *in situ* and modeled chl-*a* ($p > 0.05$) were shown as dash.

		Model	Calibration R^2	RMSE (μgL^{-1})	RMSLE (μgL^{-1})	NRMSE	MAE (μgL^{-1})	MAPE (%)	Bias
1 × 1	0 day	-	-	-	-	-	-	-	-
	±1 day	G/B	0.38	-	-	-	-	-	-
	±2 days	NIR/G	0.17	-	-	-	-	-	-
	±3 days	-	-	-	-	-	-	-	-
3 × 3	0 day	-	-	-	-	-	-	-	-
	±1 day	G/B	0.35	-	-	-	-	-	
	±2 days	G/R	0.21	-	-	-	-	-	
	±3 days	G/B	0.09	-	-	-	-	-	-
5 × 5	0 day	-	-	-	-	-	-	-	-
	±1 day	G/B	0.27	-	-	-	-	-	-
	±2 days	G/R	0.29	-	-	-	-	-	-
	±3 days	-	-	-	-	-	-	-	-

References

1. Ritchie, J.C.; Cooper, C.M.; Schiebe, F.R. The relationship of MSS and TM digital data with suspended sediments, chlorophyll, and temperature in Moon Lake, Mississippi. *Remote Sens. Environ.* **1990**, *33*(2), 137-148. doi: https://doi.org/10.1016/0034-4257(90)90039-O.
2. Lathrop, R.G.; Lillesand, T.M. Use of Thematic Mapper data to assess water quality in Green Bay and central Lake Michigan. *PE&RS*, **1986**, 52(5), 671-680.
3. Östlund, C.; Flink, P.; Strömbeck, N.; Pierson, D.; Lindell, T. Mapping of the water quality of Lake Erken, Sweden, from imaging spectrometry and Landsat Thematic Mapper. *Sci. Total Environ.,* **2001**, 268(1-3), 139-154. doi: https://doi.org/10.1016/S0048-9697(00)00683-5.
4. Tyler, A.N.; Svab, E.; Preston, T.; Présing, M.; Kovács, W.A. Remote sensing of the water quality of shallow lakes: A mixture modelling approach to quantifying phytoplankton in water characterized by high-suspended sediment. *Int. J. of Remote Sens.* **2006**, 27(8), 1521-1537. doi: https://doi.org/10.1080/01431160500419311.
5. Allan, M.G.; Hamilton, D.P.; Hicks, B.J.; Brabyn, L. 2011. Landsat remote sensing of chlorophyll a concentrations in central North Island lakes of New Zealand. *Int. J. Remote Sens.* **2011**, 32(7), 2037-2055. doi: https://doi.org/10.1080/01431161003645840.
6. Allan, M.G.; Hamilton, D.P.; Hicks, B.; Brabyn, L. Empirical and semi-analytical chlorophyll a algorithms for multi-temporal monitoring of New Zealand lakes using Landsat. *Environ. Monit. Assess.* **2015**, 187, 1-24. doi: https://doi.org/10.1007/s10661-015-4585-4.
7. Sudheer, K.P.; Chaubey, I.; Garg, V. Lake water quality assessment from landsat thematic mapper data using neural network: an approach to optimal band combination selection1. *J. Am. Water Resour. Assoc.* **2006**, 42(6), 1683-1695. doi: https://doi.org/10.1111/j.1752-1688.2006.tb06029.x.
8. Han, L.; Jordan, K.J. Estimating and mapping chlorophyll-a concentration in Pensacola Bay, Florida using Landsat ETM+ data. *Int. J. Remote Sens.* **2005**, 26(23), 5245-5254. doi: https://doi.org/10.1080/01431160500219182.
9. Doxaran, D.; Froidefond, J.M.; Castaing, P.; Babin, M. Dynamics of the turbidity maximum zone in a macrotidal estuary (the Gironde, France): Observations from field and MODIS satellite data. *Estuar. Coast. Shelf Sci.* **2009**, 81(3), 321-332. doi: https://doi.org/10.1016/j.ecss.2008.11.013.
10. Hellweger, F.L.; Miller, W.; Oshodi, K.S. Mapping turbidity in the Charles River, Boston using a high-resolution satellite. *Environ. Monit. Assess.* **2007**, 132, 311-320. doi: https://doi.org/10.1007/s10661-006-9535-8.
11. Sass, G.Z.; Creed, I.F.; Bayley, S.E.; Devito, K.J. Understanding variation in trophic status of lakes on the Boreal Plain: A 20-year retrospective using Landsat TM imagery. *Remote Sens. Environ.* **2007**, 109(2), 127–141, doi: 10.1016/j.rse.2006.12.010.
12. Floricioiu, D.; Rott, H.; Rott, E.; Dokulil, M.; Defrancesco, C. 2003. Retrieval of limnological parameters of perialpine lakes by means of MERIS data. In Proceedings of

the **2004** Envisat & ERS Symposium (ESA SP-572), 6–10 September 2004, Salzberg, Austria, pp. 1–5 (Paris: ESA).
13. Strömbeck, N.; Candiani, G.; Giardino, C.; Zilioli, E. Water quality monitoring of Lake Garda using multi-temporal MERIS data. In Proceedings of MERIS User Workshop (ESA SP-549), 10–13 November **2003**, Frascati, Italy, p. 17.1 (Paris: ESA).
14. Tebbs, E.J.; Remedios, J.J.; Harper, D.M. (2013). Remote sensing of chlorophyll-a as a measure of cyanobacterial biomass in Lake Bogoria, a hypertrophic, saline- alkaline, flamingo lake, using Landsat ETM. *Remote Sens. Environ*. **2013**, 135, 92-106. doi:10.1016/j.rse.2013.03.024.
15. Duan, H.; Zhang, Y.; Zhang, B.; Song, K.; Wang, Z. Assessment of chlorophyll-a concentration and trophic state for Lake Chagan using Landsat TM and field spectral data. *Environ. Monit. Assess*. **2007**, 129(1), 295- 308. doi:10.1007/s10661-006-9362-y
16. Mishra, S.; Mishra, D.R. Normalized difference chlorophyll index: A novel model for remote estimation of chlorophyll-a concentration in turbid productive waters. *Remote Sens Environ* **2012,** 117, 394-406. doi: 10.1016/j.rse.2011.10.016.
17. Zhengjun, W.; Jianming, H.; Guisen, D. Use of satellite imagery to assess the trophic state of Miyun Reservoir, Beijing, China. *Environ. Pollut*. **2008**, 155(1), 13-19. doi: https://doi.org/10.1016/j.envpol.2007.11.003.
18. Alawadi, F. Detection of surface algal blooms using the newly developed algorithm surface algal bloom index (SABI). In Remote Sensing of the Ocean, Sea Ice, and Large Water Regions **2010** (Vol. 7825, p. 782506). International Society for Optics and Photonics. doi: https://doi.org/10.1117/12.862096.
19. Brivio, P.A.; Giardino, C.; Zilioli, E. 2001. Determination of chlorophyll concentration changes in Lake Garda using an image-based radiative transfer code for Landsat TM images. *Int. J. Remote Sens.* **2001**, 22(2-3), 487-502. doi: https://doi.org/10.1080/014311601450059.
20. Kabbara, N.; Benkhelil, J.; Awad, M.; Barale, V. Monitoring water quality in the coastal area of Tripoli (Lebanon) using high-resolution satellite data. *ISPRS J. Photogramm. Remote Sens.* **2008**, 63(5), 488-495. doi: https://doi.org/10.1016/j.isprsjprs.2008.01.004.

```
// ==================================================
// DSWE 2.0 Monthly Composites
// ==================================================
// This script creates monthly composites of Dynamic Surface Water Extent (DSWE) v2.0
products:
// categories of ground surface inundation as detected in cloud-/shadow-/snow-free
// Landsat pixels. Where multiple values are returned for pixels within a month,
// higher-confidence water categories take precedence over lower/partial ones, while water
// or no-water categories dominate cloudy pixels. Algorithm and approach were originally
// derived from JJones's 2015 DSWE Algorithm Description pdf and updated to match changes
in
// June 2018 DSWE v2.0 documentation.
// -------------------------------------------------------------------
// DSWE coding: Jessica J. Walker, Roy E. Petrakis, Christopher E. Soulard
// -------------------------------------------------------------------
// Input: User-supplied date range (within Landsat TM availability)
// Output: Single multi-band image of monthly DSWE composites as a GEE Asset
// -------------------------------------------------------------------
// DSWE categories:
// 0 - Not Water
// 1 - Water - High Confidence
// 2 - Water - Moderate Confidence
// 3 - Partial Surface Water Pixel
// 4 - Water or wetland, low confidence
// 9 - Cloud, Cloud Shadow, or Snow (Hillshaded pixels set to 9 instead of 0)
// null - Fill (no data) ** currently left masked
// -------------------------------------------------------------------
// Notes:
// This script uses the WGS 84 projection (EPSG:4326)
// -------------------------------------------------------------------

// ---------------- Input user-required info --------------------------
// Import variables (Cambodia shapefile imported as asset for publication)
// Can also import study area or create area in GEE and save as a feature collection
//Var geometry: Table "import extent variable"
// Load DEM file, SRTM used here
```

```
var dem = ee.Image('USGS/SRTMGL1_003') //Image "SRTM Digital Elevation Data 30m"
// Id: USGS/SRTMGL1_003
// Dates should be within Landsat TM range (Aug 22, 1982 to present)
var startdate = ee.Date('1988-01-01');
var enddate = ee.Date('2018-12-31');
// -----------------------------------------------------------------------
// -----------------------------------------------------------------------
// Define AOI (var geometry)
var aoi = ee.FeatureCollection(geometry);
// -----------------------------------------------------------------------
// Load Landsat imagery
// -----------------------------------------------------------------------
// Define Landsat surface reflectance bands
var sensor_band_dict = ee.Dictionary({
                    l8 : ee.List([1,2,3,4,5,6,10]),
                    l7 : ee.List([0,1,2,3,4,6,9]),
                    l5 : ee.List([0,1,2,3,4,6,9]),
                    l4 : ee.List([0,1,2,3,4,6,9])
                    });
// Sensor band names corresponding to selected band numbers
var bandNames = ee.List(['blue','green','red','nir','swir1','swir2','pixel_qa']);
// -----------------------------------------------------
// Landsat 4 - Data availability Aug 22, 1982 - Dec 14, 1993
var ls4 = ee.ImageCollection('LANDSAT/LT04/C01/T1_SR')
            .filterBounds(aoi.geometry())
            .select(sensor_band_dict.get('l4'), bandNames);

// -----------------------------------------------------
// Landsat 5 - Data availability Jan 1, 1984 - May 5, 2012
var ls5 = ee.ImageCollection('LANDSAT/LT05/C01/T1_SR')
            .filterBounds(aoi.geometry())
            .select(sensor_band_dict.get('l5'), bandNames);
// Landsat 7 data are only used during operational SLC and
// to fill the gap between the end of LS5 and the beginning
// of LS8 data collection
// Prior to SLC-off
```

```
// ------------------------------------------------------
// Landsat 7 - Data availability Jan 1, 1999 - Aug 9, 2016
// SLC-off after 31 May 2003
var ls7 = ee.ImageCollection('LANDSAT/LE07/C01/T1_SR')
            .filterDate('1999-01-01', '2003-05-31')
            .filterBounds(aoi.geometry())
            .select(sensor_band_dict.get('l7'), bandNames);
// Post SLC-off; fill the LS 5 gap
// ------------------------------------------------------
// Landsat 7 - Data availability Jan 1, 1999 - Aug 9, 2016
// SLC-off after 31 May 2003
var ls7_2 = ee.ImageCollection('LANDSAT/LE07/C01/T1_SR')
            .filterDate('2012-05-05', '2014-04-11')
            .filterBounds(aoi.geometry())
            .select(sensor_band_dict.get('l7'), bandNames);
// -------------------------------------------------------
// Landsat 8 - Data availability Apr 11, 2014 - present
var ls8 = ee.ImageCollection('LANDSAT/LC08/C01/T1_SR')
            .filterBounds(aoi.geometry())
            .select(sensor_band_dict.get('l8'), bandNames);

// Merge landsat collections
var l4578 = ee.ImageCollection(ls4
            .merge(ls5)
            .merge(ls7)
            .merge(ls7_2)
            .merge(ls8).sort('system:time_start'))
            .filterDate(startdate, enddate);
// ---------------------------------------------------------------------
// Mask clouds, cloud shadows, and snow
// ---------------------------------------------------------------------
// https://landsat.usgs.gov/sites/default/files/documents/ledaps_product_guide.pdf
function maskClouds(img) {
  var qa = img.select(['pixel_qa']);
  var clouds = qa.bitwiseAnd(8).neq(0).or // Cloud shadow (0 = clear, 1 = contamination)
            (qa.bitwiseAnd(16).neq(0)).or // Snow
```

```
            (qa.bitwiseAnd(32).neq(0)); // Cloud
  return img.addBands(clouds.rename('clouds')); // Add band of contaminated pixels
}
// Apply mask
var img_masked = l4578.map(maskClouds); //.map(function(img){return img.clip(aoi)});
// -------------------------------------------------------------------
// Calculate hillshade mask
// -------------------------------------------------------------------
function addHillshade(img) {
   var solar_azimuth = img.get('SOLAR_AZIMUTH_ANGLE');
   var solar_zenith = img.get('SOLAR_ZENITH_ANGLE'); // solar altitude = 90-zenith
   var solar_altitude = ee.Number(90).subtract(ee.Number(solar_zenith));
  return img.addBands(ee.Terrain.hillshade(dem, solar_azimuth,
solar_altitude).rename('hillshade'));
}
// Add hillshade bands
var img_hillshade = img_masked.map(addHillshade);
// -------------------------------------------------------------------
// Calculate DSWE indices
// -------------------------------------------------------------------
function addIndices(img){
// NDVI
   img = img.addBands(img.normalizedDifference(['nir', 'red']).select([0], ['ndvi']));
// MNDWI (Modified Normalized Difference Wetness Index) = (Green - SWIR1) / (Green +
SWIR1)
   img = img.addBands(img.normalizedDifference(['green', 'swir1']).select([0], ['mndwi']));
// MBSRV (Multi-band Spectral Relationship Visible) = Green + Red
   img = img.addBands(img.select('green').add(img.select('red')).select([0], ['mbsrv'])).toFloat();
// MBSRN (Multi-band Spectral Relationship Near-Infrared) = NIR + SWIR1
   img = img.addBands(img.select('nir').add(img.select('swir1')).select([0], ['mbsrn']).toFloat());
// AWEsh (Automated Water Extent Shadow) = Blue + (2.5 * Green) + (-1.5 * mbsrn) + (-0.25 *
SWIR2)
   img = img.addBands(img.expression('blue + (2.5 * green) + (-1.5 * mbsrn) + (-0.25 * swir2)', {
      'blue': img.select('blue'),
      'green': img.select('green'),
      'mbsrn': img.select('mbsrn'),
```

```
        'swir2': img.select('swir2')
    }).select([0], ['awesh'])).toFloat();
    return img;
}
// Add indices
var img_indices = img_hillshade.map(addIndices);
// ------------------------------------------------------------------
// DSWE parameter testing
// ------------------------------------------------------------------
// Bitmask of 11111 = 16 + 8 + 4 + 2 + 1 = 31 = 1F in hex
// 1. ========== Function: test MNDWI ===========
// If (MNDWI > 0.124) set the ones digit (i.e., 00001)
function test_mndwi(img) {
  var mask = img.select('mndwi').gt(0.124);
  return img.addBands(mask
            .bitwiseAnd(0x1F)
            .rename('mndwi_bit'));
}
// 2. ======== Function: compare MBSRV and MBSRN ========
// If (MBSRV > MBSRN) set the tens digit (i.e., 00010)
function test_mbsrv_mbsrn(img) {
  var mask = img.select('mbsrv').gt(img.select('mbsrn'));
  return img.addBands(mask
            .bitwiseAnd(0x1F)
            .leftShift(1)  // shift left 1 space
            .rename('mbsrn_bit'));
}
// 3. ======== Function: test AWEsh ========
// If (AWEsh > 0.0) set the hundreds digit (i.e., 00100)
function test_awesh(img) {
  var mask = img.select('awesh').gt(0.0);
  return img.addBands(mask
             .bitwiseAnd(0x1F)
             .leftShift(2)  // shift left 2 spaces
             .rename('awesh_bit'));
}
```

```
// 4. ======= Function: test PSW1 ========
// If (MNDWI > -0.44 && SWIR1 < 900 && NIR < 1500 & NDVI < 0.7) set the thousands digit
(i.e., 01000)
function test_mndwi_swir1_nir(img) {
  var mask = img.select('mndwi').gt(-0.44)
            .and(img.select('swir1').lt(900))
            .and(img.select('nir').lt(1500))
            .and(img.select('ndvi').lt(0.7));
  return img.addBands(mask
           .bitwiseAnd(0x1F)
           .leftShift(3)  // shift left 3 spaces
           .rename('swir1_bit'));
}
// 5. ======= Function: test PSW2 =========
// If (MNDWI > -0.5 && SWIR1 < 3000 && SWIR2 < 1000 && NIR < 2500 && Blue < 1000) set the
ten-thousands digit (i.e., 10000)
function test_mndwi_swir2_nir(img){
  var mask = img.select('mndwi').gt(-0.5)
            .and(img.select('swir1').lt(3000))
            .and(img.select('swir2').lt(1000))
            .and(img.select('nir').lt(2500))
            .and(img.select('blue').lt(1000));
  return img.addBands(mask
            .bitwiseAnd(0x1F)
            .leftShift(4)  // shift left 4 spaces
            .rename('swir2_bit'));
}
// Add all bitwise bands to image collection
img_indices_bit = ee.ImageCollection(img_indices)
            .map(test_mndwi)
            .map(test_mbsrv_mbsrn)
            .map(test_awesh)
            .map(test_mndwi_swir1_nir)
            .map(test_mndwi_swir2_nir);
// Function: consolidate individual bit bands
function sum_bit_bands(img){
```

```
  var bands = img.select(['mndwi_bit', 'mbsrn_bit', 'awesh_bit', 'swir1_bit', 'swir2_bit']);
  var summed_bands = bands.reduce(ee.Reducer.bitwiseOr());
  return img.addBands(summed_bands.rename('summed_bit_band'));
}
// Add individual bit bands to image collection and summarize
var img_indices_bit = ee.ImageCollection(img_indices)
            .map(test_mndwi)
            .map(test_mbsrv_mbsrn)
            .map(test_awesh)
            .map(test_mndwi_swir1_nir)
            .map(test_mndwi_swir2_nir)
            .map(sum_bit_bands);
// ----------------------------------------------------------------------
// Produce DSWE layers
// ----------------------------------------------------------------------
// Construct slope image from DEM
//var dem = dem.clip(aoi); // removed clipping in an attempt to speed up script
var slope = ee.Terrain.slope(dem);
// Convert binary code into 4 DSWE categories
var img_indices_all = img_indices_bit.map(function(img){
  var reclass = img.select('summed_bit_band').remap([0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
                                    10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
                                    20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
                                    30, 31],

                                    [0, 0, 0, 4, 0, 4, 4, 2, 0, 4,
                                     4, 2, 4, 2, 2, 1, 4, 4, 4, 2,
                                     4, 2, 2, 1, 3, 2, 2, 1, 2, 1,
                                     1, 1]).rename('dswe');
// ID cloud-contaminated pixels
  reclass = reclass.where(img.select('clouds').eq(1), 9);
// ID shaded areas
  reclass = reclass.where(img.select('hillshade').lte(110), 8);
// ID slopes
  reclass = reclass.where(img.select('dswe').eq(4) && slope.gte(5.71).or  // 10% slope = 5.71°
            (img.select('dswe').eq(3) && slope.gte(11.31)).or           // 20% slope = 11.31°
```

```
        (img.select('dswe').eq(2) && slope.gte(16.7)).or          // 30% slope = 16.7°
        (img.select('dswe').eq(1) && slope.gte(16.7)), 0);        // 30% slope = 16.7°

  return img.addBands(reclass).select('dswe');
});
// -------------------------------------------------------------------
// Create monthly composites
// -------------------------------------------------------------------
// Generate sequence of months
var months_diff = enddate.difference(startdate, 'month'); // # of months between start and end
var months = ee.List.sequence(0, months_diff.int());
// Standardize start date to first of that month; i.e., 08-13-2000 --> 08-01-2000
var start_year = startdate.get('year');
var start_month = startdate.get('month');
var start_date = ee.Date.fromYMD(start_year, start_month, 1);
// Cycle through all months. Using a "min" reducer means that higher-confidence
// water categories take precedence over lower/partial ones, while water or no-water categories
// dominate cloudy pixels. Adding an empty band is necessary because GEE triggers
// an error if an image can't be returned, and in some months there are no
// images due to the gap between Landsat 5 and 8.
var dswe_ic = ee.ImageCollection.fromImages(months.map(function(m){
  var start = start_date.advance(m, 'month'); // Advance start date by m months
  var end = start.advance(1, 'month');        // End date is start month +1
  return img_indices_all.filterDate(start, end)
       .select(['dswe'])
       .reduce(ee.Reducer.min()) //  Produces band 'dswe_min'
       .set('date', start.format("YYYYMM")) // Sets yr/mo as property 'date'
       .cast({'dswe': 'uint8'})  // Adds empty band 'dswe'
       .remap([0,1,2,3,4,8,9], [0,1,2,3,4,9,9]).rename('dswe'); // remap takes the 1st band unless
otherwise
                                    // specified. If 'dswe_min' exists, that's the
                                    // 1st one. If 'dswe' is the only band, it is
                                    // converted to the same type as true min bands
                                    // but is empty. In either case, a band is
                                    // produced.
}));
```

```
// -------------------------------------------------------------------
// Transform image collection to multi-band image for output
// -------------------------------------------------------------------
function appendBand(current, previous){
  // Build a name for the band (here, "dswe_YYYYMM")
  var bandName =
ee.Algorithms.String(current.bandNames().get(0)).cat('_').cat(current.get('date'));
  // Rename the band
  current = current.select([0], [bandName]);
  // Append it to the result (only returns current item on first element/iteration)
  var accum = ee.Algorithms.If(ee.Algorithms.IsEqual(previous, null), current,
current.addBands(ee.Image(previous)));
  // Return the accumulation
  return accum;
}
// Iterate through the image collection
var dswe_i = ee.Image(dswe_ic.iterate(appendBand));
// -------------------------------------------------------------------
// Export image
// -------------------------------------------------------------------
  Export.image.toDrive({
    image: dswe_i,
    description: 'DSWE2_1988_2018_Cambodia',
        scale: 30,
    crs: 'EPSG:4326',  // WGS 84
    maxPixels: 1e13,
    region: aoi.geometry().bounds(),
  });
// -------------------------------------------------------------------
// Visualization of DSWE Image Products
// -------------------------------------------------------------------
// Center on polygon
Map.centerObject(aoi, 10);
Map.addLayer(aoi, {}, 'aoi');
// Viz parameters: classes: 0, 1, 2, 3, 4, 9
```

```
var dswe_viz = {min:0, max: 9, palette: ['000000', '002ba1', '6287ec', '77b800', 'c1bdb6', '000000', '000000',
                              '000000', '000000', 'ffffff']};
var ls_viz =  {bands: [ 'red', 'green', 'blue'], min:0, max:3000, gamma: [0.95, 1.1, 1]};
//var dswe_ic_firstImage = ee.Image(dswe_ic.first());
//Map.addLayer(dswe_ic_firstImage, dswe_viz_params_all, 'dswe_ic_firstImage');

// DSWE image (not composited)
Map.addLayer(ee.Image(img_indices_all.select('dswe').first()), dswe_viz, "DSWE");
// DSWE monthly composite image (from image collection)
Map.addLayer(dswe_ic.first().select('dswe'), dswe_viz, "DSWE composite");
// Landsat Data
Map.addLayer(ee.Image(img_indices.first()), ls_viz, 'Landsat');
```